Semi-Annual Report on

Computer-Aided Circuit Analysis

Submitted to

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Office of Grants and Research Contracts

Washington D. C. 20546

This work was done under the NASA grant NGR -39-023-004, during the period May 15 to November 14, 1965, at the Electrical Engineering Department, Villanova University Villanova, Pennsylvania.

PRINCIPAL INVESTIGATOR

Tsute Yang

TSUTE YANG
Professor of Electrical Engineering

Henry T. Koonce

HENRY T. KOONCE
Director of Research and Development

RESEARCH AND DEVELOPMENT DIVISION

VILLANOVA UNIVERSITY

VILLANOVA, PENNSYLVANIA

Semi-Annual Report of the Research on

Digital Computer-Aided Circuit Analysis

under the NASA Grant NGR-39-023-004

covering period from May 15 to November 14, 1965.

I. Literature Search

One of the initial activities of computer-aided analysis of electric circuits consisted of a literature search in the subject area. Publications in the professional and technical journals may be classified into two catagories: those dealing with the general approach of analyzing circuit performance of any configuration within given sets of constraints, and those tailored for the design of specific types of networks. The primary interest of this project has been in the investigation and development of techniques and programs for the analytic solution of circuit problems in general. There is evidence of growing interest and activity both in industry and research institutions in exploiting digital computers as an aid in system design and reliability evaluation. It gradually comes to the realization that progress and practicality of the concepts involves more than the mathematical model formulation and digital algorithm applied to circuit solutions. Various other facets such as man-machine interface, time cost of operation, etc. will influence and contribute significantly to the success of the program.

Three existing digital computer programs written expressly for circuit analysis and evaluation were reviewed and examined, namely, the Automated Statistical Analysis Program (ASAP), the Circuit Analysis System (CIRCS), and the Electronic Circuit Analysis Program (ECAP); the first and the third being developed by the International Business Machine Corporation, and the second at the Jet Propulsion Laboratory. They may be regarded as the offspring of the same lineage, because they share the same philosophy of attacking the problem and they

possess strong similarities in modeling and formating. All three programs are capable of accepting a topological description of the circuit in simple language, writing the circuit equations according to Kirchhoff's current law, and carrying out the analysis requested.

The ASAP is primarily designed to perform a Monte Carlo statistical analysis on the d-c currents and voltages of circuits containing transistors and diodes. It computes two types of sensitivities. The first type is a qualitative analysis where the measure of the spread of each parameter about the mean value is taken into consideration. The second type is based on a one per cent deviation of each component parameter from its nominal value. The CIRCS program provides options of d-c, a-c, and transient analysis, and also the Mandex worst case and sensitivity calculations. The ECAP, which has recently been released to general public, has the additional feature of including mutual inductance as a circuit element without finding its equivalent tee or pi. The ASAP works on the IBM-7090/94 computer while the other two operate on IBM-1620 with a 1311 disk file system. CIRCS requires a 20k core storage unit while ECAP requires a 40k core storage. Because ECAP is inclusive of the features of CIRCS, discussions and observations will be made in the next section of the report with regard to ASAP and ECAP only.

A comprehensive bibliography of current literature published in connection with computer-aided circuit analysis and design has been in preparation and is expected to be completed and distributed in a few weeks.

II. Study of Available Programs -- ASAP and ECAP

One of the justifications in using the digital computer for circuit analysis and design is to obtain information concerning the circuit operation and performance which would otherwise be unobtainable by other means, either for physical reasons or for time considerations. For instance, in the reliability study of a circuit comprising

many component parts, it is practically impossible to find out systematically all the effects on the output of varying each component to a different extent on a lab bench. However, a well conceived computer program will have the fortet of carrying out the simulation faithfully and exhaustively. It is with this objective that the ASAP uses Monte Carlo method to produce various statistics of the circuit voltages and currents for any assigned range of tolerance and any shape of statistical distribution curve for each circuit element.

Diodes and transistors in the circuit are to be specified by piecewise linear v-i curves for the diodes, by I_b-V_{be} and I_c-V_{ce} curves for the transistor. There may be 2 to 10 values for each curve. The program determines the equivalent circuit for the diode or transistor and an iterative procedure is followed in locating the operating point. The automatic computation requires a large amount of input data and computer time. Moreover, the convergence of the process in arriving at a satisfactory operating point may be difficult to realize.

ASAP, in writing the nodal equations from the topological description in the data input, uses a pattern recognition subroutine to produce a trace table and establishes the algebraic equations satisfying Kirchhoff's current law. It is significant that the equations are solved algebraically in symbolic form by the Gauss reduction method without back-substitution. The back-substitution occurs numerically during the execution phase. It is quite probable that, during the solution process, some intermediate equation may become longer than the alloted storage space. This may arise as a result of the complexity of the circuit or of the particular sequence of solving one unknown after another. It would be an important factor which could severly limit the actual size of the circuit which can be handled by the program. The official statement concerning the capability limits of the ASAP lists 50 dependent nodes (a dependent node is defined as any node other than ground or those connected

to a voltage source) and 40 diodes plus transistors. If these figures truly represent the upper limit of the program, it seems that ASAP will be found useful in quite large population of circuit configurations in practice.

The ASAP program requires a relatively large machine configuration to operate, which may not be readily available in some circumstances. Designers are hoping to be able to make use of digital computers as compactly as a cathode-ray oscilloscope, if not demanding the comparable size and elementary simplicity in use as a slide rule. Technology will advance and meet the challenge in time. At the present time, however, efforts are made in developing programs adaptable for small size computer operation. The ECAP is such an undertaking. The complete ECAP program can be obtained through the IBM 1620 Users' Group.

The ECAP is a card input program designed for operation on IBM 1620 with 1311 disk storage drive. It has the features of automatic equation writing, three options of analysis, d-c, a-c, and transient, computation of partial derivatives and sensitivity coefficients of voltages, and automatic logarithmic modification of frequency in the a-c analysis portion.

Transistors and diodes are represented by their equivalent circuits in the analysis. In the transient calculations, the parameter values of the diode and transistor can be made to vary as a function of circuit voltages and currents. To accomplish this, the complement of the circuit elements which are recognized by ECAP contains a "switch" element, which presents the pertinent equivalent circuit for a particular operating region. Thus the three commonly referred to regions of operation of a transistor, cutoff, active, and saturation, can be handled adequately in a similar manner that diodes can be conducting with different forward resistance or nonconducting.

In the ECAP program the sensitivity coeffidents are defined and calculated only for node voltages as their change for a 1% change in the branch parameter. In the worst case analysis both worst-case maximum and worst-case minimum are computed. In the former calculation, positive partial derivatives are multiplied by positive tolerances and negative partial derivatives by negative tolerances. In the latter, positive partial derivatives are multiplied by negative tolerances, etc. The basic assumption is that the circuit output variables are linearly related to the parameter values. This approximation is valid when the parameter tolerances are small. When the tolerance exceeds 10% of the nominal value, the manual recommends the parameter substitution method. First, the partial derivatives of the node voltages are calculated. Then the nominal value of each parameter in the circuit is replaced with its maximum or minimum value, in accordance with the sign of the corresponding partial derivative, and the result is treated as a new ECAP job.

In the d-c analysis program, the d-c parameter modification solutions for a given circuit are obtained by correcting the nodal impedance matrix or the equivalent current vector associated with the circuit. This imposes the condition that the tolerance has to be limited in range in the automatic determination of worst cases. However, the a-c modification solutions, on the other hand, are completely new. Consequently it allows any extent of parameter change in the calculation.

A maximum of five coupled inductances can be included in the circuit that is to be analyzed. This is a feature not often found in other programs.

The transient response of node voltages and element currents are produced by ECAP at the start of a transient solution and at uniform intervals of time thereafter, until the end of the solution is reached. In addition, these output variables are also produced immediately before and immediately after each switch actuation, if any. The time of the switch actuation is also given.

The system of integro-differential equations which arises in the transient analysis is solved in ECAP by an implicit numerical integration technique. It involves two main tasks: the replacement of the system of integro-differential equations by an equivalent set of algebraic difference equations, and the repetitive numerical solution of the algebraic equations. In solving the equations at the end of each series of discrete intervals of time, each new solution is dependent upon the results of the previous solution. That is, the values of certain of the terms in the set of algebraic equations are always computed from the results of the previous solution. For the first solution (at the end of the first time step) these terms are evaluated from the circuit initial conditions. Therefore, the results of each solution become the initial conditions of the succeeding solutions.

III Adaption of Current Techniques of Digital Computer-Aided Circuit Analysis to Moderate Size Computers.

General - One of the areas of interest to the project for investigation is the possible use of the small computer for circuit analysis. Since the IBM 1620 digital computer* is a small machine and is available on campus, it was decided to study programming methods that could be performed using this computer. One method that seemed particularly suitable for programming by the IBM 1620 is the scheme used on the British general purpose computer called Deuce [1]. This method will give the solution of driving point and transfer functions of cascaded networks as a function of sinusoidal frequencies.

The Deuce method of programming was selected for the following reasons: 1) many practical circuits consist of cascaded stages with simple network geometry, although the circuits are composed of many components, 2) it permits and encourages the circuit designer to analyze his design with a minimum of programming experience

* IBM 1620 Model 1, with 60K memory positions.

in a span of time commensurate with bread-boarding a circuit, 3) the program can easily be modified to cope with configurations of various complexities and 4) the program can be run by the designer on a small computer.

The Analysis Problem - The Deuce type program consists essentially of determining the basic structure of each section of a cascaded network which will be used in the analysis, and of interpreting the code to be assigned to the elements of the structure. As an illustration of determining the basic structure, consider the network shown in fig. 1.

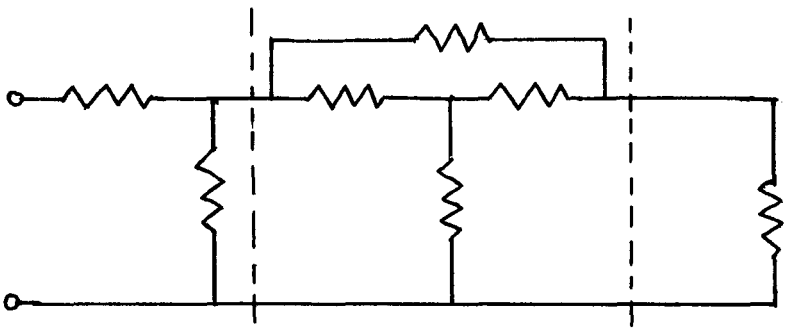

Fig. 1 A Cascaded Bridged-T With Degenerate Sections.

Since one section of the network is the bridged-T, the network must be considered a tandem connection of three bridged-T sections, two of which have branches missing. Once the basic structure of the sections is decided, the driving point and transfer functions are easily derived. A table of these functions for all the common network configurations can be made and used as needed in the programs. The program analysis proceeds step by step beginning with the output terminals of the network until the input terminals are reached as shown in fig. 2.

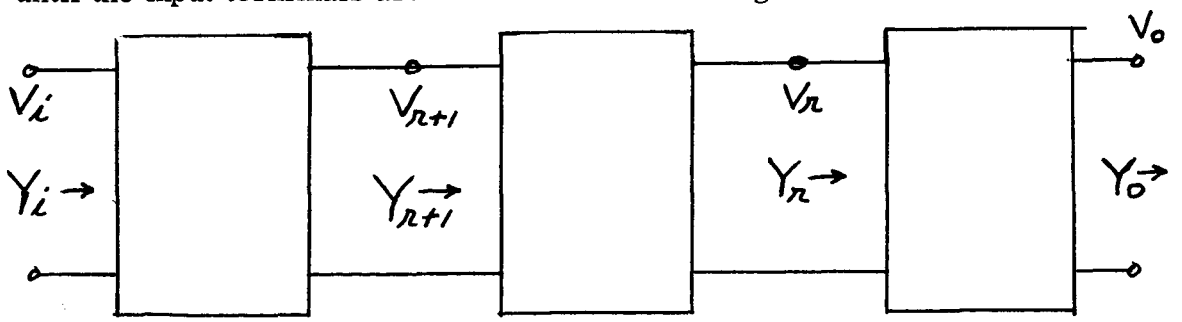

Fig. 2 Cascaded Network For Analysis.

Each section is analyzed knowing the output voltage and output admittance. The output voltage V_o of the last section is assumed to be 1.0 volts and Y_o is assumed to be 0 mhos.** The input voltage V_r and the admittance Y_r of one section are calculated using the appropriate equations. Thus with V_r and Y_r known, V_r+1 and Y_r+1 of the next section are calculated. This procedure is continued until V_i and Y_i are determined.

The above concepts will be illustrated by the following simple example of a resistive ladder network shown in fig. 3, in which the sections are composed of an inverted L network. The problem is to find V_o/V_i and Y_i. The procedure is as follows:

1) The equations for the solution of the voltage transfer function and driving point admittance per section of the ladder network is derived as:

$$V_r = V_o \ [1 + (G_r + G_o) R_r]$$

$$Y_r = V_o / V_r \ (G_r + G_o)$$

where G_o is the load admittance

G_r is the shunt admittance

R_r is the series resistance.

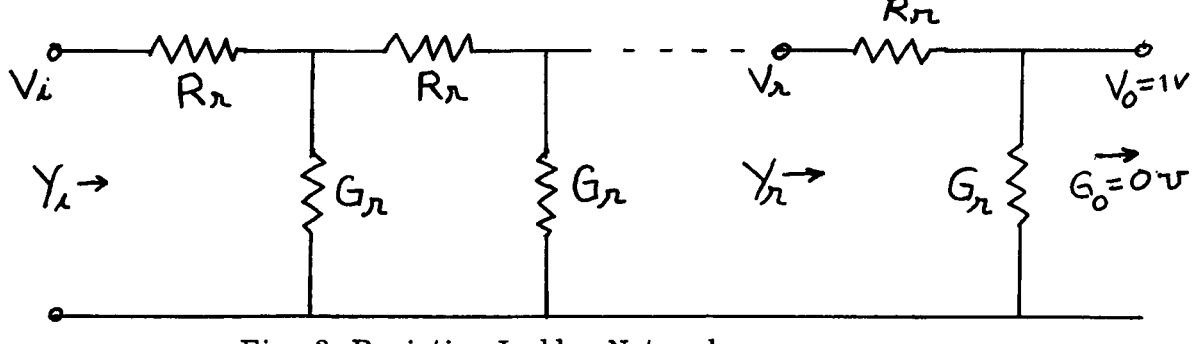

Fig. 3 Resistive Ladder Network.

** Note that although V_o is assumed equal to one volt, the actual value of V_i applied to the circuit will probably change this value, but it will not affect the value of the computed transfer function. Similarly Y_o may be other than zero but this is simply specified at the start of the program, before Y_i is computed.

2) Code each resistance to indicate if it is to be considered as a series or shunt component.

3) Compute V_r and equate it to V_o for the next section.

4) Compute Y_r and equate it to G_o for the next section.

5) repeat steps 3 and 4 until V_i and Y_i are determined.

See Fig. 4 for the flow chart used to program this problem.

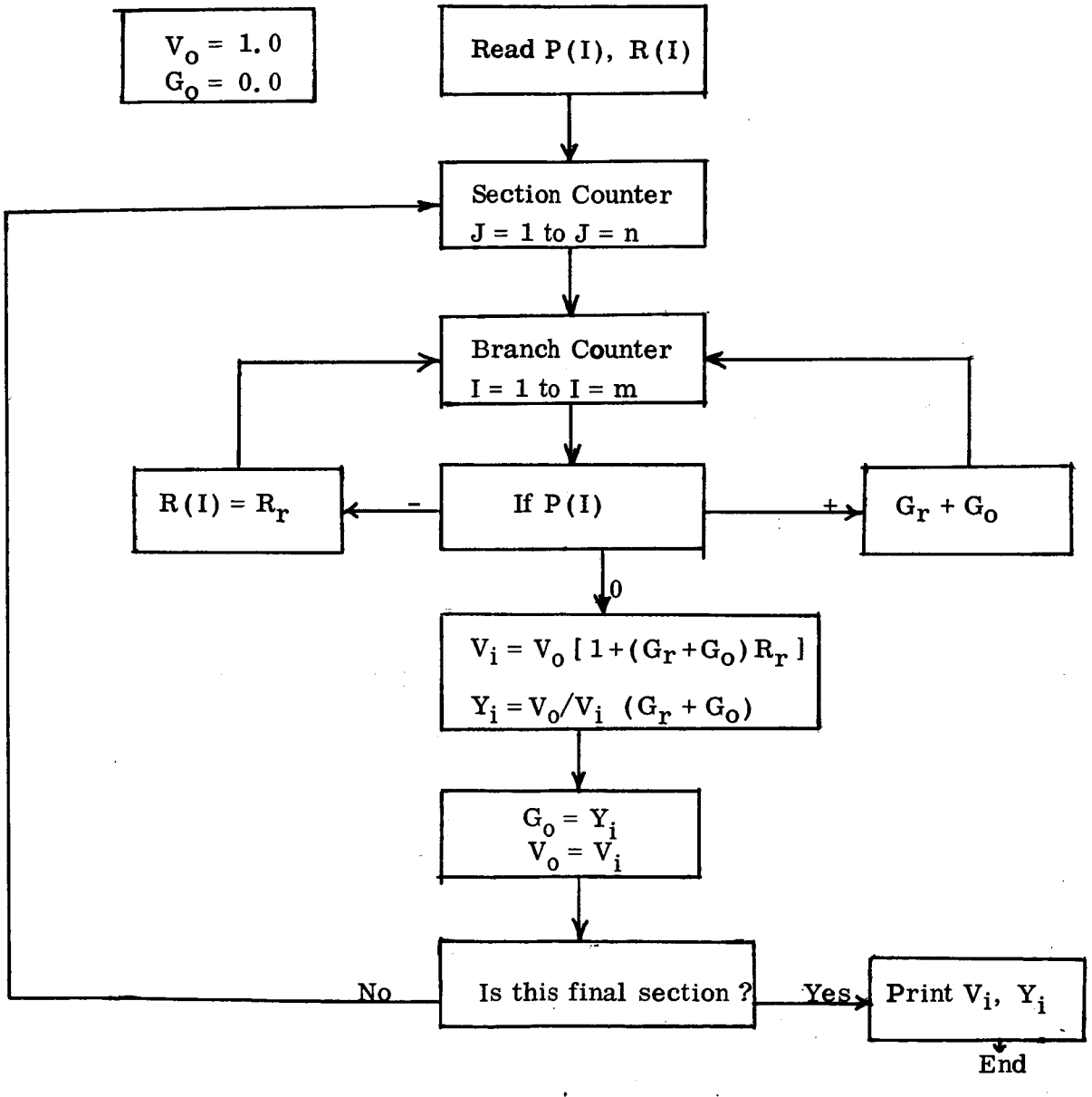

Fig. 4 Flow chart for solution of resistive ladder network.

Coding The Input Data - In general each IBM card is divided into a number of fields. See Fig. 5. Two fields F(I) and G(I) are used to designate the position of the network branch in the general structure of the cascaded section. A branch is here defined to consist of at most a series RLC circuit. Three other fields are used to indicate the value of the L, C, and R components. Note that the type and value of the component is designated by giving it a value in a specific position in the fields. This alone determines that it is L, C, or R. In actual use the symbol S, (where S = 1/C) has been used instead of C since values of infinite C cannot be processed by the computer, and the symbol H instead of L since L represents a number without a decimal in Fortran programming. If H, S, or R are short circuits, the value of zero is entered into the respective field. See card No. 6 in fig. 5. Since the computer cannot process fractions divisible by zero the program must test for these values and accomodate their results in the overall solution.

Using symbols F(I) and G(I) to code the HSR branch in the structure permits nine different combinations of F(I) and G(I) when using the Fortran IF statement. See Table 1, for an explanation of the coding that was used to analyze the two stage RC coupled network shown in fig. 6. This circuit was analyzed

Card No.	F(I)	G(I)	H(I)	S(I)	R(I)
1	-1.0	-2.0	0.0	0.0	10^3
2	0.0	0.0	0.0	10^7	0.0
3	1.0	0.0	0.0	0.0	10^3
4	1.0	1.0	20.0	0.0	0.0
5	1.0	1.0	0.0	0.0	10^5
6	1.0	1.0	0.0	10^7	0.0
7	1.0	1.0	0.0	0.0	0.0

Fig. 5 IBM card with code and data of each branch.

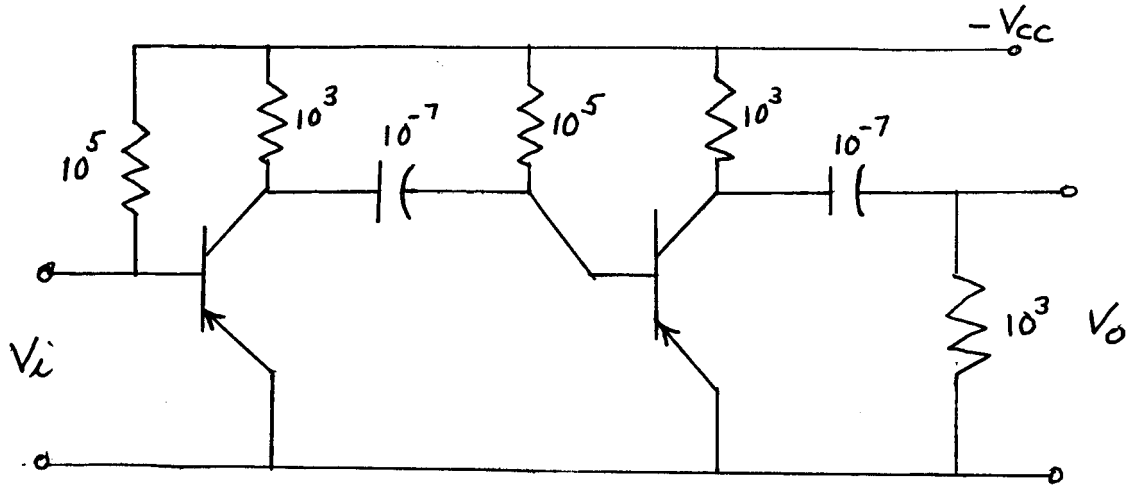

Fig. 6a Two Stage RC Coupled Amplifier.

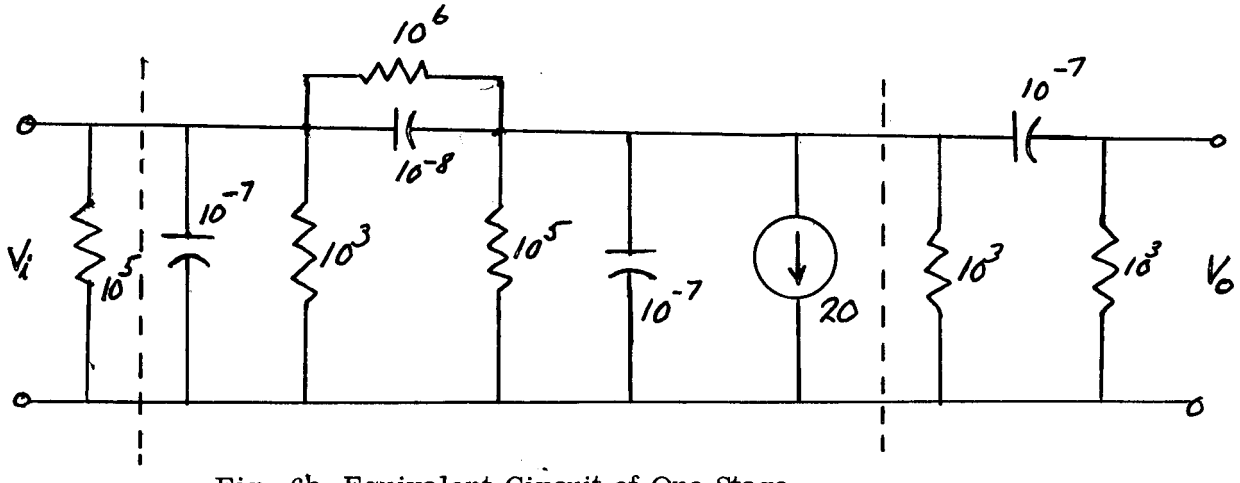

Fig. 6b Equivalent Circuit of One Stage.

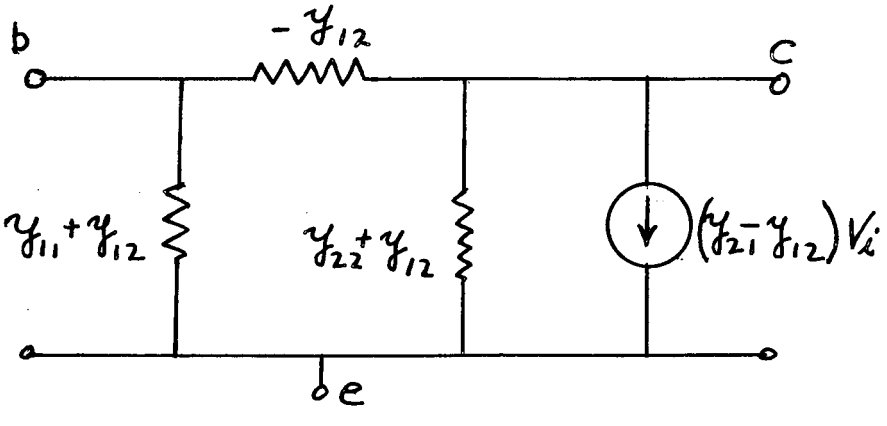

Fig. 6c Transistor Equivalent Circuit.

TABLE 1

F(I)	G(I)	
1	0	Indicates a branch that is part of a parallel circuit.
1	1	Indicates a current source G, or value H(I) and angle S(I).
1	2	Indicates a voltage source E, of value H(I) and angle S(I).
-1	-1	Indicates a series branch that is in series with a parallel circuit, both of which are in the impedance part of the structure.
-1	-1	Indicates a series branch that is in series with a parallel circuit, both of which are in the admittance part of the structure.
-1	-2	Indicates only one branch exists in an admittance part of the structure.
0	0	Indicates only one branch exists in an impedance part of the structure.
0	-1	not presently used
0	1	not presently used

TABLE 2

$W/_{rps}$	V (volts)	Θ_{V_i} (rps)	Y_i (mhos)	Θ_{Y_i} (rps)
1	50.55×10^{-6}	-2.712	43.95	0.4284
1,000	51.6×10^{-6}	-2.21	43.84	0.4249
10,000	11.6×10^{-6}	0.00156	43.11	0.4280
10^5	10.5×10^{-4}	1.68	43.08	0.436
10^6	10.4×10^{-3}	-3.27	18.29	1.51

for V_i and Y_i varying frequently from one radian per second to 10^6 rps. See Table 2 for the results of this study.

The transistor short circuit admittance model, shown in fig. 6 was used because it fit into the structure of the ladder network. Values for its equivalent circuit were assumed, although the program could easily be modified to process the short-circuit admittance parameters.

Concluding Remarks - The study of the Deuce method of programming has answered some questions and generated new ones to be answered. Some of the answered questions are: 1) how does one handle complex numbers on the computer? This was solved by the method of vertical and horizontal components, 2) what is the length of time and the size of the memory needed to solve a cascaded network? The machine time for the solution of the two stage RC coupled amplifier was approximately two and one-half minutes, which includes compiling and loading times. Memory space occupied by the program was about thirty thousand positions, and 3) how much programming experience is necessary to write or modify a program of circuit analysis of the Deuce type? Judging by the experience gained on the project, anyone who can design circuits can program a cascaded network analysis. While the programmer might not be as efficient as an experienced programmer, this is not considered of much importance when using a small computer.

Some of the problems generated from the study of the Deuce method are: 1) is it possible when using this method to determine if transistor and diode circuits are operating in the ON or OFF state? At this time it does not seem feasible to predict this since the information regarding the device parameters must be specified in order to calculate the responses, 2) can a time domain response be calculated using the Deuce method? A method of evaluating the time domain response from the complex frequency response has been presented by M. S. Corrington [2].

This technique works for solutions involving small amounts of time. A program was written and the results verified the method. But the processing of information using the Deuce method will require the use of the computer in performing non-numerical algebra. Investigations along this line are presently being studied and indications are that it can and has been done (3). Future work in this area is being contemplated, 3) will the Deuce method lend itself to statistical studies of cascaded networks? Some thought has been given to this problem, and successful efforts have been made in making operational a random number generator subroutine for possible Monte Carlo studies.

References:

1. E. A. Pacello: "The Use of Deuce for Network Analysis", The Marconi Review, Vol. XXIV, No. 142, p.p. 101-114 1961.

2. M. S. Corrington: "Simplified Calculation of Transient Response" Proceeding of the IEEE, vol. 53, p.p. 287-292, March 1965.

3. W. S. Brown, The ALPAK System for Non-Numerical Algebra on a Digital Computer, Part I, Bell Syst. Tech. Jour. Vol. 42, no. 5, p.p. 2081-2119, September, 1963.

   W. S. Brown, J. P. Hyde, and B. A. Tague, The ALPAK System Part II, ibid. vol. 43, no. 2, p.p. 785-804, March 1964.

   J. P. Hyde, The ALPAK System, Part III, ibid. vol. 43 no. 4 pt. 2, p.p. 1547-1562, July 1964.

IV. Future Plans

1. Continue the effort of bibliography preparation and issue supplements as necessary after its completion.

2. Outstanding in the project during the next six month period is the operational experience and critical study of the ECAP program. Because it is the only program available at the present time to cover d-c, a-c, and transient analysis on a moderate size machine, it seems justifiable to find the merits and shortcomings of the program especially with respect to the computer time requirement and ease of interpretation and usefulness of the output data.

3. Negotiation is under way with the General Electric Co. to use the STANPAK program developed by their Computer Division. Recently a time sharing line has been installed between the Space Technology Center of General Electric Co. at King of Prussia and Villanova University campus, which gives Villanova direct access to the GE computer complex. As much as is known before the program is released, STANPAK is basically a reliability prediction and tolerance analysis program using the statistical approach. It handles d-c circuits only and lacks facilities for a-c and transient computations. It would be useful nevertheless to examine its structure and operation as a contemporary of ASAP.